6.2 Proofs with Number Theory

In Chapter 3, we studied how to express statements precisely using the language of predicate logic. But just as English enables us to make both true and false claims, the language of predicate logic allows for the expression of both true and false sentences. In this chapter, we will turn our attention to analyzing and communicating the truth or falsehood of these statements. You will develop the skills required to answer the following questions:

- How can you figure out if a given statement is True or False?
- If you know a statement is True, how can you convince others that it is True? How can you do the same if you know the statement is False instead?
- If someone gives you an explanation of why a statement is True, how do you know whether to believe them or not?

These questions draw a distinction between the internal and external components of mathematical reasoning. When given a new statement, you'll first need to figure out for yourself whether it is true (internal), and then be able to express your thought process to others (external). But even though we make a separation, these two processes are certainly connected: it is only after convincing yourself that a statement is true that you should then try to convince others. And often in the process of formalizing your intuition for others, you notice an error or gap in your reasoning that causes you to revisit your intuition—or make you question whether the statement is actually true!

A **mathematical proof** is how we communicate ideas about the truth or falsehood of a statement to others. There are many different philosophical ideas about what constitutes a proof, but what they all have in common is that a proof is a mode of *communication*, from the person creating the proof to the person digesting it. In this course, we will focus on reading and creating our own written mathematical proofs, which is the standard proof medium in computer science.

As with all forms of communication, the style and content of a proof varies depending on the audience. In this course, the audience for all of our proofs will be an average computer science student (and not your TA or instructor). As we will discuss, your audience determines how formal a proof should be (here, quite formal), and what background knowledge you can assume is understood without explanation (here, not much).

However, there is even variation in the typical computer science student with experience in this area, so as much as possible in this course, we will introduce *new mathematical domains* to serve as the objects of study in our proofs.

This approach has three very nice benefits: first, by building domains from the ground up, we can specify absolutely the common definitions and properties that everyone may assume and use freely in proofs; second, these domains are the theoretical foundation of many areas of computer science, and learning about them here will serve you well in many future courses; and third, learning about new domains will help develop the skill of *reading about a new mathematical context and understanding it*.[1] The definitions and axioms of a new domain communicate the foundation upon which we build new proofs—in order to prove things, we need to understand the objects that we're talking about first.

First examples

We're going to start out our exploration of proofs by studying a few simple statements. Our first foray into domain exploration will be into number theory, which you can think of as taking a type of entity with which we are quite familiar, and formalizing definitions and pushing the boundaries of what we actually know about these *numbers* that we use every day.

You may find our first few examples a bit on the easy side, which is fine. We are using them not so much for their ability to generate mathematical insight, but rather to model both the *thinking* and the *writing* that would go into approaching a problem.

Each example in this section is divided into three or four parts:

1. The statement that we want to prove or disprove. Sometimes, we'll specify whether to prove or disprove it, and other times deciding whether the statement is true or false is part of the exercise.
2. A translation of the statement into predicate logic. This step often provides insight into the *logical structure* of the statement that we are considering, which in turn informs the structure and techniques that we will use in our proofs.
3. A discussion to try to gain some intuition about why the statement is true. You'll tend to see that these are written very informally, as if we are talking to a friend on a whiteboard. The discussion usually will reveal the mathematical insight that forms the content of a proof. **This is often the hardest part of developing a proof, so please don't skip these sections!**
4. A formal proof. This is meant to be a standalone piece of writing, the "final product" of our earlier work. Depending on the depth of the discussion, the formal proof might end up being almost mechanical – a matter of formalizing our intuition.

With this in mind, let's dive right in!

Example. Prove that $23 \mid 115$.

[1] In other words, you won't just learn about new domains; you'll learn *how* to learn about new domains!

Translation. We will *expand* the definition of divisibility to rewrite this statement in terms of simpler operations:

$$\exists k \in \mathbb{Z},\ 115 = 23k.$$

Discussion. We just need to divide 115 by 23, right?

Proof. Let $k = 5$.

Then $115 = 23 \cdot 5 = 23 \cdot k$.[2]

We can draw from this example a more general technique for structuring our existence proofs. A statement of the form $\exists x \in S,\ P(x)$ is True when at least one element of S satisfies P (hence our use of `any` in Python). The easiest way to convince someone that this is True is to actually find the concrete element that satisfies P, and then show that it does.[3] This is so natural a strategy that it should not be surprising that there is a "standard proof format" when dealing with such statements.

A typical proof of an existential.

Given statement to prove: $\exists x \in S,\ P(x)$.

Proof. Let $x =$ ________.

[Proof that $P($________$)$ is True.] ■

Note that the two blanks represent the same element of S, which *you* get to choose as a prover. Thus existence proofs usually come down to *finding* a correct element of the domain which satisfy the required properties.

Here is another example which uses the same idea, but with two existentially-quantified variables.

Example. Prove that there exists an integer that divides 104.

Translation. There is the key phrase "there exists" right in the problem statement, so we could write $\exists a \in \mathbb{Z},\ a \mid 104$. We can once again expand the definition of divisibility to write:[4]

[2] We typically signal the end of a proof by writing a black square ■ in the bottom-right corner.

[3] Of course, this is *not* the only proof technique used for existence proofs. You'll study more sophisticated ways of doing such proofs in future courses.

$$\exists a, k \in \mathbb{Z},\ 104 = ak.$$

Discussion. Basically, we need to pick a pair of divisors of 104. Since this is an existential proof and we get to pick both a and k, any pair of divisors will work.

Proof. Let $a = -2$ and let $k = -52$.

Then $104 = ak$.

■

The previous example is the first one that had multiple quantifiers. In our proof, we had to give explicit values for both a and k to show that the statement held. Just as how a *sentence* in predicate logic must have all its variables quantified, a *mathematical proof* must introduce all variables contained in the sentence being proven.

Alternating quantifiers, revisited

In the Chapter 3, we saw how changing the order of an existential and universal quantifier changed the meaning of a statement. Now, we'll study how the order of quantifiers changes how we can introduce variables in a proof.

Example. Prove that all integers are divisible by 1.

Translation. The statement contains a universal quantification: $\forall n \in \mathbb{Z},\ 1 \mid n$. We can unpack the definition of divisibility to

$$\forall n \in \mathbb{Z},\ \exists k \in \mathbb{Z},\ n = 1 \cdot k.$$

Discussion. The final equation in the fully-expanded form of the statement is straightforward, and is valid when k equals n. But how should I introduce these variables? Answer: *in the same order they are quantified in the statement.*

Proof. Let $n \in \mathbb{Z}$. Let $k = n$.

Then $n = 1 \cdot n = 1 \cdot k$.

■

This proof is quite short, but introduces a few new elements. First, it introduced a variable n that could represent any real number. Unlike the previous existence proofs, when we introduced this variable n we did not specify a concrete value like 10, but rather said that n was an arbitrary real number by writing ``Let $n \in \mathbb{Z}$.[5]

[4] We use the abbreviated form for two quantifications of the same type.

[5] You might notice that we use the same word "let" to introduce both existentially- and universally-quantified variables. However, you should always be able to tell how the variable is quantified based on whether it is

given a concrete value or an “arbitrary” value in the proof.

A typical proof of a universal.

Given statement to prove: $\forall x \in S,\ P(x)$.

Proof. Let $x \in S$. (That is, let x be an arbitrary element of S.)

[Proof that $P(x)$ is True]. ∎

The other interesting element of this proof was that it contained an existentially-quantified variable k after the $\forall n \in \mathbb{Z}$. We used an extremely important tool at our disposal when it comes to proofs with multiple quantifiers: **any existentially-quantified variable can be assigned a value that depends on the variables defined before it.**

In our proof, we first defined n to be an arbitrary integer. Immediately after this, we wanted to show that for this n, $\exists k \in \mathbb{Z},\ n = 1 \cdot k$. And to prove this, we needed a value for k—a “let” statement. Because we define k after having defined n, we can use n in the definition of k and say “Let $k = n$.” It may be helpful to think about the analogous process in programming. We first initialize a variable n, and then define a new variable k that is assigned the value of n.

Even though this may seem obvious, one important thing to note is that the *order of variables in the statement determines the order in which the variables must be introduced in the proof*, and hence which variables can depend on which other variables. For example, consider the following erroneous “proof.”

Example. (Wrong!) Prove that
$\exists k \in \mathbb{Z},\ \forall n \in \mathbb{Z},\ n = 1 \cdot k$.

Proof. Let $k = n$. Let $n \in \mathbb{Z}$.

Then $n = 1 \cdot k$. ∎

This proof may look very similar to the previous one, but it contains one crucial difference. The very first sentence, “Let $k = n$,” is invalid: at that point, n has not yet been defined![6]

[6] This is analagous to a `NameError` in Python.

This is the result of having switched around the order of the quantifiers, which forces k to be defined independently of whatever n is chosen.

Note: don’t assume that just because *one* proof is invalid, that *all* proofs of this statement are invalid! We cannot conclude that this statement is False just because we found one proof that didn't work.[7] That said, this statement is indeed False, and we'll see later on in this chapter how to prove that a statement is False instead of True.

Proofs involving implications

Let's look at one new example.

Example. Prove that for all integers x, if x divides $(x+5)$, then x also divides 5.

Translation. There is both a universal quantification and implication in this statement:[8]

$$\forall x \in \mathbb{Z},\ x \mid (x+5) \Rightarrow x \mid 5.$$

When we unpack the definition of divisibility, we need to be careful about how the quantifiers are grouped:

$$\forall x \in \mathbb{Z},\ \big(\exists k_1 \in \mathbb{Z},\ x+5 = k_1 x\big) \Rightarrow \big(\exists k_2 \in \mathbb{Z},\ 5 = k_2 x\big).$$

Discussion. I need to prove that if x divides $x+5$, then it also divides 5. To prove this, I'm going to *assume* that x divides $x+5$, and I need to *prove* that x divides 5.

Since x is divisible by x, I should be able to subtract it from $x+5$ and keep the result a multiple of x. Can I prove that using the definition of divisibility? I basically need to "turn" the equation $x+5 = k_1 x$ into the equation $5 = k_2 x$.

Proof. Let x be an arbitrary integer. *Assume* that $x \mid (x+5)$, i.e., that there exists $k_1 \in \mathbb{Z}$ such that $x + 5 = k_1 x$. We want to prove that there exists $k_2 \in \mathbb{Z}$ such that $5 = k_2 x$.

Let $k_2 = k_1 - 1$.

Then we can calculate:

[7] A meta way of looking at this: a statement is True when *there exists* a correct proof of it.

[8] As we noted back in Chapter 3, the "universal + implication" form is the most common form of statement we encounter.

$$\begin{aligned} k_2 x &= (k_1 - 1)x \\ &= k_1 x - x \\ &= (x + 5) - x \quad \text{(we assumed } x + 5 = k_1 x\text{)} \\ &= 5 \end{aligned} \qquad \blacksquare$$

Whew, that was a bit longer than the proofs we've already done. There were a lot of new elements that we introduced here, so let's break them down:

- After introducing x, we wanted to prove the *implication* $x \mid (x + 5) \Rightarrow x \mid 5$. To prove an implication, we needed to assume that the hypothesis was True, and then prove that the conclusion is also True. In our proof, we wrote "**Assume** $x \mid (x + 5)$."

  This is *not* a claim that $x \mid (x + 5)$ is True; rather, it is a way to consider what would happen *if* $x \mid (x + 5)$ were True. The goal for the rest of the proof was to prove that $x \mid 5$.

  This proof structure is common when proving an implication:

  A typical proof of an implication (direct).

  Given statement to prove: $p \Rightarrow q$.

  Proof. Assume p.

  [Proof that q is True.] $\blacksquare$

- When we assumed that $x \mid (x + 5)$, what this really did was introduce a new variable $k_1 \in \mathbb{Z}$ from the definition of divisibility. This might seem a little odd, but take a moment to think about what this means in English. We assumed that x divides $x + 5$, which (by definition) is the same as assuming that there exists an integer k_1 such that $x + 5 = k_1 x$. Given that such a number exists, we can give it a name and refer to it in the rest of our proof.[9]

Generalizing our example

One of the most important meta-techniques in mathematical proof is that of **generalization**: taking a true statement (and a proof of the statement), and then replacing a concrete value in the statement with a universally quantified variable. For example, consider the statement from the previous example, $\forall x \in \mathbb{Z},\ x \mid (x + 5) \Rightarrow x \mid 5$. It doesn't seem like the "5" serves any special purpose; it is highly likely that it could be replaced by another number like 165, and the statement would still hold.[10]

[9] In other words, we introduced a variable into the proof through an *assumption* we made.

[10] Concretely, consider the statement $\forall x \in \mathbb{Z},\ x \mid (x + 165) \Rightarrow x \mid 165$, which is at least as plausible as the original statement with 5's.

But rather than replace the 5 with another concrete number and then re-proving the statement, we will instead replace it with a universally-quantified variable, and prove the corresponding statement. This way, we will know that in fact we could replace the 5 with *any* integer and the statement would still hold.

Example. Prove that for all $d \in \mathbb{Z}$, and for all $x \in \mathbb{Z}$, if x divides $(x + d)$, then x also divides d.

Translation. This has basically the same translation as last time, except now we have an extra variable:

$$\forall d, x \in \mathbb{Z},\ \big(\exists k_1 \in \mathbb{Z},\ x + d = k_1 x\big) \Rightarrow \big(\exists k_2 \in \mathbb{Z},\ d = k_2 x\big).$$

Discussion. I should be able to use the same set of calculations as last time.

Proof. Let d and x be arbitrary integers. *Assume* that $x \mid (x + d)$, i.e., there exists $k_1 \in \mathbb{Z}$ such that $x + d = k_1 x$. We want to prove that there exists $k_2 \in \mathbb{Z}$ such that $d = k_2 x$.

Let $k_2 = k_1 - 1$.

Then we can calculate:

$$\begin{aligned} k_2 x &= (k_1 - 1)x \\ &= k_1 x - x \\ &= (x + d) - x \\ &= d \end{aligned}$$

■

This proof is basically the same as the previous one: we have simply swapped out all of the 5's with d's. We say that the proof *did not depend on the value* 5, meaning there was no place that we used some special property of 5, where we could have used a generic integer instead. We can also say that the original statement and proof *generalize* to this second version.

Why does generalization matter? By generalizing the previous statement from being about the number 5 to an arbitrary integer, we have essentially gone from one statement being true to an infinite number of statements being true. The more general the statement, the more useful it becomes. We care about exponent laws like $a^b \cdot a^c = a^{b+c}$ precisely because they apply to every possible number; regardless of what our concrete calculation is, we know we can use this law in our calculations.